NMF-BASED KEYWORD LEARNING FROM SCARCE DATA

Bart Ons, Jort F. Gemmeke and Hugo Van hamme

Department ESAT-PSI, KULeuven, Leuven, Belgium

Bart.Ons@esat.kuleuven.be, Jort.Gemmeke@esat.kuleuven.be, Hugo.Vanhamme@esat.kuleuven.be

ABSTRACT

This research is situated in a project aimed at the development of a vocal user interface (VUI) that learns to understand its users specifically persons with a speech impairment. The vocal interface adapts to the speech of the user by learning the vocabulary from interaction examples. Word learning is implemented through weakly supervised non-negative matrix factorization (NMF). The goal of this study is to investigate how we can improve word learning when the number of interaction examples is low. We demonstrate two approaches to train NMF models on scarce data: 1) training word models using smoothed training data, and 2) training word models that strictly correspond to the grounding information derived from a few interaction examples. We found that both approaches can substantially improve word learning from scarce training data.

Index Terms— weakly supervised non-negative matrix factorization, vocabulary acquisition, vocal user interface, data scarcity

1. INTRODUCTION

Command and Control (C&C) speech recognition allows users to control different conditions in their environment like the central heating or the light units in the house, but also to interact with devices like smartphones or computers. This study is situated in the "Adaptation and Learning for Assistive Domestic Vocal Interfaces" (ALADIN) project [1, 2] aimed at the development of a Vocal User Interface (VUI) that can understand normal as well as deviant speech. The user should be able to choose his own words, phrases or sounds in order to control the target application at hand.

We meet this objective by grounding the word learning process of the VUI in the environment of the end user, so that the VUI is trained by mining the speech input from the end user and the changes that are provoked on a device. For instance, the user says: "Please, turn on the television" and turns on the television with the remote control. The learning problem is a machine learning problem where the user has to demonstrate the intended action to the VUI, and by doing so, he provides supervision to the spoken utterance [2]. The supervision for training the speech recognizer is only weak, since the changes provoked on the device, resulting for instance from a button push, cannot be transformed in an orthographic transcription with correct word order as is required in training conventional automatic speech recognition systems based on Hidden Markov Models (HMMs) [3].

As an alternative, non-negative matrix factorization (NMF) has been presented as a useful machine learning procedure to discover and learn the acoustic representation of spoken words guided by weak supervision [4, 5, 6]. In short, NMF decomposes utterance-based representations into two low-rank representations, one representing the recurring acoustic patterns such as spoken words, and one describing which recurring patterns are active in each utterance.

The goal of this study is to investigate how we can improve fast vocabulary acquisition in the state-of-the-art NMF approach [4]. Fast learning is an essential objective as it reduces the user's effort to train the system and allows faster workability of the VUI. It is achieved when word models trained on *scarce* speech data can still generalize to new speech signals. We propose two approaches to improve the word recognition rates: 1) *smoothing* of the acoustic model posterior probabilities in order to avoid over-training of the NMF word models, and 2) *restricting* the acoustic representation of the word models to correspond exactly to the supervision data, i.e. the grounding information. If a spoken word and its supervision information only appear one time, it is not a recurrent pattern and difficult to detect by NMF. By imposing the supervision, we essentially seek representations for words that appeared only once. We will evaluate the effectiveness of both approaches by doing word learning experiments with increasing amounts of training data.

The paper is organized as follows. In Section 2, we briefly explain *supervised NMF* and the processing steps to build the feature vectors for NMF, namely *soft vector quantisation* (soft VQ) [7] and the *histograms of acoustic co-occurrence (HAC)* [6]. In Section 3, we describe the two approaches to improve the generalization of the models. We conduct two experiments, one for each method and we report the results in section 4. Finally, in Section 5 and 6 we discuss the proposed methods and conclude with our conclusions and thoughts for future work.

This work is funded by IWT-SBO project 100049 (ALADIN)

2. BACKGROUND

2.1. Acoustic representation

Fixed length feature vectors are required for NMF. We build utterance-based fixed length vectors by transforming the acoustic feature vectors into a Gaussian posteriorgram [7] and by accumulating the posterior probabilities to an histogram of acoustic co-occurrence (HAC) [4, 6] .

A posteriorgram is a two dimensional data structure containing the posterior probability that a frame-based feature vector (first dimensions: time) was emitted by a particular acoustic unit (second dimension: class). Here, the classes are Gaussians obtained by k-means clustering followed by the estimation of a full covariance Gaussian based on all frame observations falling in each respective cluster [8, 7]. Each entry is the relative (normalized) likelihood that the observation was emitted from the respective Gaussian.

The posteriorgram of an utterance has a variable length that depends on the number of frames in an utterance. We create HAC features to build a fixed-length vector for each utterance and to incorporate time information. In the HAC, the probability of co-occurrence between frames, τ frames apart from each other, is accumulated over one whole utterance for all possible cluster pairs. Coarser and more fine grained code books as well as more time information are added by stacking HAC's with different time lags and different codebooks in one utterance-based vector. The vector length F depends on te number L_i of Gaussians in each codebook $i = 1, \ldots, C$ and the number of time lags T: $F = T \times \sum_{i=1}^{C} L_i^2$. The data matrix consisting of the acoustic representation of N utterances is denoted by $\mathbf{V}_a(F \times N)$ with F the number of features.

2.2. Grounding information

In addition to the acoustic representation, there is a second input stream providing utterance-based supervision denoted by $\mathbf{V}_g(K \times N)$ with K the number of words defining the demonstrated actions on a device, also called keywords. Supervision in each utterance is indicated as follows: there is one row in $\mathbf{V}_g$ for each keyword and its entries represent the number of times that the respective word was uttered in the n^{th} utterance. In the context of the VUI of Section 1, this assumes VUI actions such as pushing a button are related to one or more keywords. Supervision is weak in the sense that the absence or presence of keywords are indicated without any chronological information within the utterance.

2.3. The supervised NMF framework

2.3.1. Training

NMF [9] decomposes a data matrix $\mathbf{V}$ into the product of two lower rank matrices, $\mathbf{W}$ and $\mathbf{H}$. A variant to NMF is supervised NMF [4, 6] where supervision $\mathbf{V}_g$ is added to the data matrix $\mathbf{V}_a$ and an additional part is added in the lower rank matrix $\mathbf{W}$ to regularize the factorization in correspondence with the supervision. The model is:

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_g \\ \mathbf{V}_a \end{bmatrix} \approx \begin{bmatrix} \mathbf{W}_g \\ \mathbf{W}_a \end{bmatrix} \mathbf{H} = \mathbf{W}\mathbf{H} \tag{1}$$

with all entries in $\mathbf{V}$, $\mathbf{W}$ and $\mathbf{H}$ constrained to be non-negative.

The purpose of supervised NMF learning is to uncover the acoustic representation of each keyword. The columns in $\mathbf{W}_a$ represent the latent structure (recurring patterns) of the columns in $\mathbf{V}_a$ associated to a keyword. The columns in $\mathbf{H}$ indicate which patterns are combined to approximate the columns in $\mathbf{V}$.

When the total set of vocal commands contains K keywords, $\mathbf{W}$ should count at least K columns, but in practice, some D extra columns are added to $\mathbf{W}$ to model the filler words.

Different loss functions are possible and the most appropriate loss function depends on the statistical structure of the data matrix. An appropriate loss function for the approximation in Eq. 1 assuming that entries in $\mathbf{V}$ are counts of events is the generalised Kullback-Leibler divergence (gkld) or I-divergence:

$$D_{KL}(\mathbf{V}||\mathbf{W}\mathbf{H}) = \sum_i \sum_n \left[v_{in} \log \frac{v_{in}}{[\mathbf{W}\mathbf{H}]_{in}} - v_{in} + (\mathbf{W}\mathbf{H})_{in} \right] \tag{2}$$

with $i = 1, \ldots, I$, $I = K + F$ and $n = 1, \ldots, N$.

Lee and Seung [9] derived the following alternating multiplicative update rules for minimizing Eq. 2 as a function of the entries h_{rn} of $\mathbf{H}$ and w_{ir} of $\mathbf{W}$. Convergence is guaranteed to a local optimum:

$$h_{rn} \leftarrow h_{rn} \frac{\sum_i \frac{v_{in}}{[\mathbf{W}\mathbf{H}]_{in}} w_{ir}}{\sum_q w_{qr}} \tag{3}$$

$$w_{ir} \leftarrow w_{ir} \frac{\sum_n \frac{v_{in}}{[\mathbf{W}\mathbf{H}]_{in}} h_{rn}}{\sum_p h_{rp}} \tag{4}$$

with v_{in} entries of $\mathbf{V}$, and $r = 1, \ldots, R = K + D$ with R the inner dimension of $\mathbf{W}$ and $\mathbf{H}$. After each update of $\mathbf{W}$, we normalise its columns to sum to unity in order to prevent arbitrary scaling of $\mathbf{W}$ and $\mathbf{H}$.

In supervised NMF, the first K rows in $\mathbf{H}$ are initialized as $\mathbf{V}_g$ and the first $K \times K$ entries in $\mathbf{W}_g$ are initialized as the identity matrix [8]. A small random number is added to $\mathbf{W}_g$. The initialization procedure helps convergence to a solution with keyword representations in the first K columns of $\mathbf{W}_a$. All entries in $\mathbf{W}_a$ are randomly initialized.

The solutions for $\mathbf{H}$, $\mathbf{W}_a$ and $\mathbf{W}_g$ obtained by update rules in Eq. 3 and 4 are denoted by $\mathbf{H}^*$, $\mathbf{W}_a^*$ and $\mathbf{W}_g^*$.

2.3.2. Recognition

Keyword recognition is tested on a separate set of new utterances denoted by $\mathbf{V}_t$. $\mathbf{H}_t^*$ is found by minimizing the generalized Kullback-Leibler divergence between $\mathbf{V}_t$ and $(\mathbf{W}_a^*\mathbf{H}_t)$.

$$\mathbf{H}_t^* = \arg\min_{\mathbf{H}_t} D_{KL}(\mathbf{V}_t||\mathbf{W}_a^*\mathbf{H}_t) \quad (5)$$

The optimization problem in Eq. 5 is a convex problem as $\mathbf{W}_a^*$ is held fixed, and the solution $\mathbf{H}_t^*$ is used to provide the keyword activation matrix $\mathbf{A}$ as follows:

$$\mathbf{A} = \mathbf{W}_g^*\mathbf{H}_t^* \quad (6)$$

The higher the score in the rows of $\mathbf{A}$, the more likely that the respective keyword has appeared in the spoken test utterances.

3. PROPOSED METHODS

We propose two methods to improve the word learning from scarce training data: learning from *smoothed* data and restricting the optimalization procedure to follow the supervision, referred to as *restricted word learning*.

3.1. Smoothing

In this method, we propose to smooth the data matrix by imposing smoothness on the posteriorgrams. The smoothed posteriorgram with entries $\hat{P}_{t_i,\theta}$, with θ a Gaussian from the set Φ of Gaussians and t_i the timestamp of the respective frame, smoothing is obtained as follows:

$$\hat{P}_{t_i,\theta} = \frac{(P_{t_i,\theta})^\zeta}{\sum_{\theta\in\Phi}(P_{t_i,\theta})^\zeta} \quad (7)$$

with the exponent $0 < \zeta < 1$ leading to smoother (flatter) posterior probabilities.

We investigated the effect of smoothing for small and large training sets using two different smoothing conditions: we smoothed the training data and the test data in the first condition while we only smoothed the training data in the second condition. If smoothing is helpful in reducing noise and irrelevant small-scale features, we expect an improvement in the first case over all training set sizes. The improvements gained by smoothing are then essentially depending on the data. However, if better performance is also obtained by smoothing the training data but not the test data and only for small data sets, a strong indication is provided that the smoothing of scarce data is able to provide word models that generalize better to new instances.

3.2. Restricted word learning

In this method, we keep the first K rows of $\mathbf{H}$ fixed during the multiplicative optimization updates (see Eq. 3 and 4). The first K rows of $\mathbf{H}$ correspond to the supervision and indicate the occurrence of a keyword by a number 0 or 1. However, keeping the entries in $\mathbf{H}$ fixed to 0 or 1 is actually suboptimal as a value different from 1 allows us to model the duration of the spoken words. Longer words are spread over more acoustic frames, and therefore, they have larger acoustic co-occurence counts. Since the keyword representations modelled by the first K colums of $\mathbf{W}$ are all normalized to sum to unity, the differences in word length can only be reflected in the entries of $\mathbf{H}$. Nevertheless, the supervision data in $\mathbf{V}_g$ is a good initial approximation of the optimal solution to $\mathbf{H}$ and keeping this values fixed to this initial approximation is not going to harm the optimalization process too much while we gain by reducing the dimensionality of the optimization problem. We therefore expect better word models by restricting the optimization of $\mathbf{H}$ for small data sets but not for large data sets. We call this approach "restricted word learning".

4. EXPERIMENTS

4.1. Introduction

We evaluate potential gains for the use of smoothing and restricted word learning as explained in Section 3. In the first experiment, we implemented seven smoothing values for ζ in Eq. 7, $\zeta = 0.025, 0.05, 0.1, 0.2, 0.4, 0.6$ and 1. The baseline is given by a value of 1-smoothing. We implemented the two smoothing conditions explained in Section 3.1 and evaluated smoothing for small and large training sets, $N = 50, 100, 200$ and 1785.

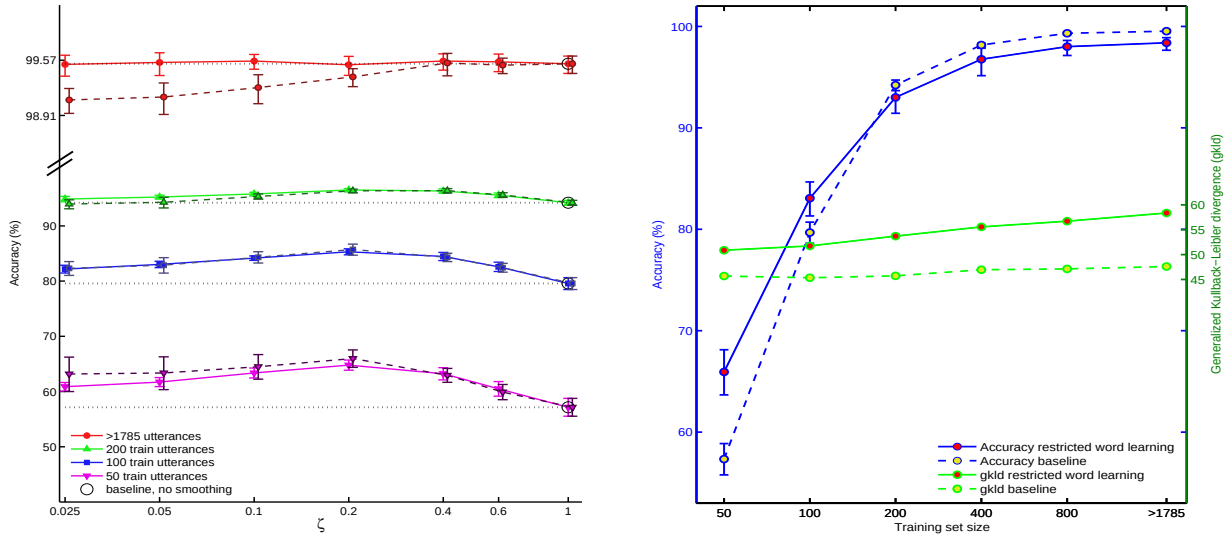
In the second experiment, we investigated six training set sizes, $N = 50, 100, 200, 400, 800$ and 1785 utterances against two different multiplicative update schemes. In the baseline condition, we used the traditional update rules as expressed in Eq. 3 and 4. In the restricted condition, we used a different update rule for $\mathbf{H}$ as explained in Section 3.2. The performance is evaluated by the accuracy expressed as the percentage of correct recognized keywords.

4.2. Experimental setup

4.2.1. Speech material

To mimic a usage situation in which no speech material of a user is (yet) available when training the word learning system, code book training is carried out on a different database than the one used for keyword learning. This means the low-level acoustic model is speaker-independent and the recording conditions differ from the user environment. We used the "Wall Street Journal corpus recorded at the University of Cambridge, phase 0", WSJCAM0 [10] for this purpose, which is the UK English equivalent of a subset of the US English Wall street Journal corpus (WSJ0).

For keyword learning we used the UK English subset of the ACORNS corpus [11] developed in the second year



(a) smoothings for different training set sizes

(b) restricted word learning

Fig. 1. *Accuracy for smoothing and restricted word learning. The error bars denote the* 95% *confidence intervals. The dashed lines in (a) are accuracies against different smoothing values used to smooth the training data while the test data is not smoothed. The solid lines in (a) are accuracies against smoothing values used to smooth both sets, training and test data. The horizontal dotted lines indicate the respective baseline performance. The blue lines in (b) are the accuracies against different training set sizes using common NMF updates (the dashed line) or restricted word learning (the solid line) corresponding with values on the Y-axis on the left. The green lines in (b) depict the generalized Kullback-Leibler divergence (gkld, see Eq. 2) between the predicted occurence of words in the test set and the plain truth*

of the ACORNS project and we selected the four speakers with the most recorded utterances. The test sets counted $593, 594, 596$ and 599 utterances for the four respective speakers and we used training sets of increasing sizes with $N = 1790, 1786, 1789$ or 1791 utterances for the largest training set for the four speakers, respectively. In ACORNS, utterances consist of 1 to 4 different keywords embedded in a carrier sentence with unrelated filler words. In total, there are 50 unique predefined keywords and 30 filler words. The choice for the corpus fit well for the purpose of evaluating the performance of the VUI since the supervision is weak (a bag of words) and the size and complexity of the data is similar to a common home automation task.

4.2.2. Features

Feature extraction was done by using Mutual Information Discriminant Analysis or MIDA [12], MIDA features consist of a linear combination of 22 log-MEL spectral dimensions and their first and second order differences (Δ and $\Delta\Delta$). The linear combination is aimed at maximizing the mutual information between the MIDA features and phone classes. The MIDA transformation was learned using the corpus WSJ-CAM0.

We used three code books of dimension $L = 20, 100$ and 400. Each code word corresponded to one Gaussian and posteriorgrams were created using the procedure described in Section 2.1.

HAC representations were created as explained in Section 2.1 using three frame lags, $\tau = 2, 5$ and 9. For each combination of frame lag and code book, there is one posteriorgram per utterance. For each utterance we obtained one fixed-length vector with the dimensionality determined by the number of code books, their sizes and the number of frame lags: $F = 3 \times (20^2 + 100^2 + 400^2) = 511200$ features for each utterance, however, feature vectors are very *sparse*.

4.2.3. Implementation

In addition to the initialisation procedure explained in Section 2.3, $D = 25$ was chosen for both experiments. Preliminary experiments showed that 100 iterations are sufficient to reach convergence. We applied five different initialisations of each respective combination of smoothing, training set size, speaker and update scheme. One possible problem could be that for $N \leq K + D$ (i.e., the training set size $N = 50$), the rank $\mathbf{W}$ is larger than the rank of $\mathbf{V}$. However, the non-negativity constraints and the supervision in NMF inhibit a trivial solution.

4.3. Results

The resulting accuracies are shown in Figure 1. For each method, there is a graph showing the average keyword recog-

nition accuracy as a function of smoothing, see 1(a), and as a function of set size for restricted and unrestricted word learning, see 1(b). The error bars denote the 95% confidence interval after controlling the variation due to the speaker variability using the procedure described in [13].

4.3.1. Smoothing

We found significant improvements with respect to the baseline (horizontal dotted lines in 1(a)) for almost all levels of smoothing (solid lines in 1(a)) when using small training sets, with $N = 50$, $N = 100$ and $N = 200$. In the smallest training set, $N = 50$, every keyword was spoken at least one time. We were able to obtain a baseline accuracy of 57% and improved the result to 66%, an improvement of 8% with respect to the baseline. However, we did not find a significant improvement for the same smoothings in the largest training set $N > 1785$.

Smoothing the training set but not the test set (dashed lines in 1(a)) shows a tendency to improve the accuracy even more for the smallest data set $N = 50$. However, the opposite trend is seen for the largest training set size. The dashed line depicting the accuracy for the largest training set in Figure 1(a) declines for more smoothing.

Clearly, smoothing behaves differently for small and large data sets and smoothing improves accuracy for scarce training data.

4.3.2. Restricted word learning

For small data sets, $N = 50$ and $N = 100$, we found a significant improvement in accuracy with respect to the baseline (the dashed lines in Figure 1(b)) by restricting word learning (the solid lines in Figure 1(b)) as explained in Section 3.2. However, for larger data sets, ($N \geq 200$), the opposite effect is displayed favoring common optimization update rules as expressed in Eq. 3 and 4. For the smallest training set, counting 50 utterances, we have an average baseline accuracy of 57% and we obtained an accuracy of 66% by restricted word learning, an improvement of 8%, quite similar to the effect of smoothing. However, for the largest dataset, $N > 1785$, the baseline (see Figure 1(b)) gave an accuracy of 99.5% while restricted word models led to a lower accuracy of 98.7%. Restricted word models are only helpful in the case of scarce data. The cost function of the test set, i.e. of $\mathbf{A}$ in Eq. 6 are depicted in a green color in Figure 1(b). Although restricted word learning allows for a better prediction of word occurence in the test set, these predictions have a higher generalized Kullback-Leibler divergence compared with the baseline.

5. DISCUSSION

Clearly, there is a relation between the amount of training data that is available and the improvements by either smoothing and restricted word learning. In general, both techniques are helpful if the number of training examples is low. The different effects of both techniques on small and large training sets demonstrate that optimization issues and model training pose a distinct challenge for scarce data. To the best of our knowledge, this distinction has not been given a lot of attention in the literature.

5.1. Smoothing

Smoothing appears to be effective for all but the largest data sets. Moreover, the optimal parameter value for smoothing is independent of the size of the training set. This can be understood as follows. Smoothing the probabilities of acoustic events causes more overlap of the Gaussians in the feature space. Without smoothing, only one or two Gaussians contribute significantly to the total probability mass of an observation as most observations lie close to the centre of a single high-dimensional Gaussian. The effect of smoothing is that observations are described by multiple Gaussians and a larger mass in their posteriorgram is shared if they are located in the same region of the feature space. As the shared probability mass between different observations corresponding to the same keyword label increases, it becomes easier to detect a recurrent pattern in the case of scarce data.

If training and test sets are smoothed, smoothing also increases the robustness of the feature representations. The training-test mismatch makes their position in the feature space uncertain within some neighbourhood. A small shift in position will affect the non-smoothed posteriorgram much more than the smoothed posteriorgram. Smoothing therefore reduces the noise level of the observation at the cost of some fine-scale resolution. A coarser but more robust representation is especially helpful for the case of scarce training data. However, for large training sets, when test sets are not smoothed, a coarser resolution of the training set affect the performance negatively as the training-test mismatch becomes larger.

A third positive effect of smoothing is related to the use of the KLD divergence. Probabilities which are underestimated during training on scarce data may have a detrimental effect during testing because of the singularities at zero and the asymmetry of the KLD. Such features have a unreasonably large impact of the total value of the cost function. The use of smoothing increases those probabilities and generally balances the impact of the acoustic features.

5.2. Restricted word models

By imposing a solution in favour of the supervision introduced in $\mathbf{H}$ (see Section 2.3), we find an adequately repre-

sentation of the keywords for which supervision is provided, i.e. the first K columns in $\mathbf{W}$, but it raises questions about the representation of the filler words for which no supervision is available, the D remaining columns in $\mathbf{W}$. The presence of filler words is randomly initialized in $\mathbf{H}$ and unsupervised learning of the filler words is solely based on detecting recurrent acoustic patterns. If filler words are adequatly represented, they are helpful for keyword recognition because they separate irrelevant patterns from relevant keyword patterns in the utterance-based representation (a bag of features). This does raise the question of whether *any* number of garbage columns ($D > 0$) can be beneficial for scarce training data, but this is left as future work.

Although better results are obtained for restricted word learning if the number of training examples is low, these better results are accompanied with a higher divergence. In different words, normal update rules learn better to minimize the generalised Kullback-Leibler divergence than the proposed approach, but keyword recognition accuracy is lower. This observation suggests that modifications to the objective function taking into account the availability of the training data and the mathematical expression of the supervision could lead to better solutions.

6. CONCLUSION

We demonstrated two techniques, smoothing and restricted word learning, to improve weakly supervised NMF learning on scarce training data. Smoothing was shown to be an effective method to substantially accelerate word learning on small data sets while maintaining the good accuracies on larger training sets. These findings are valuable since they showed that optimization issues and model training pose a distinct challenge if the availability of data is limited. Moreover, the second technique, restricted word learning seemed to improve the generalisation of the model to new data as the word models closely follow the supervision in the data.

Future research will focus on a more in-depth analysis of combining supervised and unsupervised word learning and different sorts of divergences, combining this with the proposed smoothing technique and applying smoothing to posteriorgrams obtained from alternative acoustic models such as phone recognizers and neural networks.

7. REFERENCES

[1] J. van de Loo, J. F. Gemmeke, G. De Pauw, J. Driesen, H. Van hamme, and W. Daelemans, "Towards a self-learning assistive vocal interface: Vocabulary and grammar learning," in *Proc. of the workshop Speech and Multimodal Interaction in Assistive Environments (SMIAE)*, 2012.

[2] J. F. Gemmeke, J. van de Loo, G. De Pauw, J. Driesen, H. Van hamme, and W. Daelemans, "A self-learning assistive vocal interface based on vocabulary learning and grammar induction," in *Proc. INTERSPEECH*, 2012, pp. 1–4.

[3] I. A. Clemente, M. Heckmann, and B. Wrede, "Incremental word learning: Efficient hmm initialization and large margin discriminative adaptation," *Speech Communication*, vol. 54, no. 9, pp. 1029–1048, 2012.

[4] J. Driesen, J.F. Gemmeke, and H. Van hamme, "Weakly supervised keyword learning using sparse representations of speech," in *Proc. ICASSP*, Kyoto, Japan, 2012, pp. 5145–5148.

[5] L. ten Bosch, J. Driesen, H. Van hamme, and L. Boves, "On a computational model for language acquisition: modeling cross-speaker generalisation," in *Text, Speech and Dialogue*. Springer, 2009, pp. 315–322.

[6] H. Van hamme, "Hac-models: a novel approach to continuous speech recognition," in *Proc. Interspeech*, Brisbane, Australia, 2008, pp. 255–258.

[7] M. Sun and H. Van hamme, "A two-layer non-negative matrix factorization model for vocabulary discovery," in *In ICML'11 Symposium on Machine Learning in Speech and Language Processing*, Bellevue, Washington, USA, 2011.

[8] J. Driesen, *Discovering words in speech using matrix factorization*, Ph.D. thesis, K.U.Leuven, ESAT, July 2012.

[9] D. Seung and L Lee, "Algorithms for non-negative matrix factorization," *Advances in neural information processing systems*, vol. 13, pp. 556–562, 2001.

[10] T. Robinson, J. Fransen, D. Pye, J. Foote, and S. Renals, "Wsjcam0: A british english speech corpus for large vocabulary continuous speech recognition," in *In Proc. ICASSP*, Detroit, Michigan, USA, 1995.

[11] L. Boves, L. ten Bosch, and R. Moore, "Acorns-towards computational modeling of communication and recognition skills," in *Proc. IEEE int. Conf. On Cognitive informatics*, California, USA, 2007, pp. 349–355.

[12] K. Demuynck, *Extracting, Modelling and Combining Information in Speech Recognition*, Ph.D. thesis, K.U.Leuven, ESAT, February 2001.

[13] D. Cousineau, "Confidence intervals in within-subject designs: A simpler solution to loftus and masson's method," *Tutorial in Quantitative Methods for Psychology*, vol. 1, no. 1, pp. 42–45, 2005.